

ROB316 - PLANIFICATION ET CONTRÔLE

TP n°2 Commande avec Anticipation, Commande Prédictive

12 janvier 2020

Gabriel Henrique Riqueti

Victor Kenichi Nascimento Kobayashi

ENSTA IP Paris

Résumé

Ce cours a pour objectif expliquer la commande prédictive (MPC), est une commande avec multiples entrées et sorties qui choisi la commande telle que la prédiction de l'état suive une consigne pour un coût minimal et en respectant des contraintes.

Description du TP

Le TP2 est composé par trois exercices. Le premier veut montrer l'influence de l'anticipation sur la performance du contrôle. Ensuite, on a un exercice pour vérifier la zone de stabilité d'une commande prédictive et, finalement, un exercice pour écrire et essayer la commande prédictive. Ces deux derniers questions suivent l'article [H. Chen, 1998] en utilisent le même modèle et la théorie développés dans l'article.

Question 1 - Anticipation

Pour réaliser l'approche définit d'anticipation, on a débuté par la discrétisation du chemin avec un pas égal au pas du TP1 de 0,49 m.

Ensuite, on a choisi le point à suivre par rapport en prenant en compte la fenêtre d'anticipation et le pas de discrétisation. Finalement, on a écrit la loi de contrôle.

Dans le TP précédent, on a utilisé un pas de discrétisation de 0,49 m, ce que nous a donné la Figure 1. Comme le pas actuellement utilisé est plus petit que le précédent, même avec un fenêtre d'anticipation de 5 points la distance actuelle entre le robot qu'on suppose être sur un point et le point de destin est de 0,05 m. Alors avec cette fenêtre et ce pas de discrétisation le robot poursuit un point plus proche qu'avant.

En regardant la loi de contrôle, la commande est plus vite pour poursuivre les points plus loin de la position du robot. Ainsi, la performance du chemin réalisé avec anticipation est

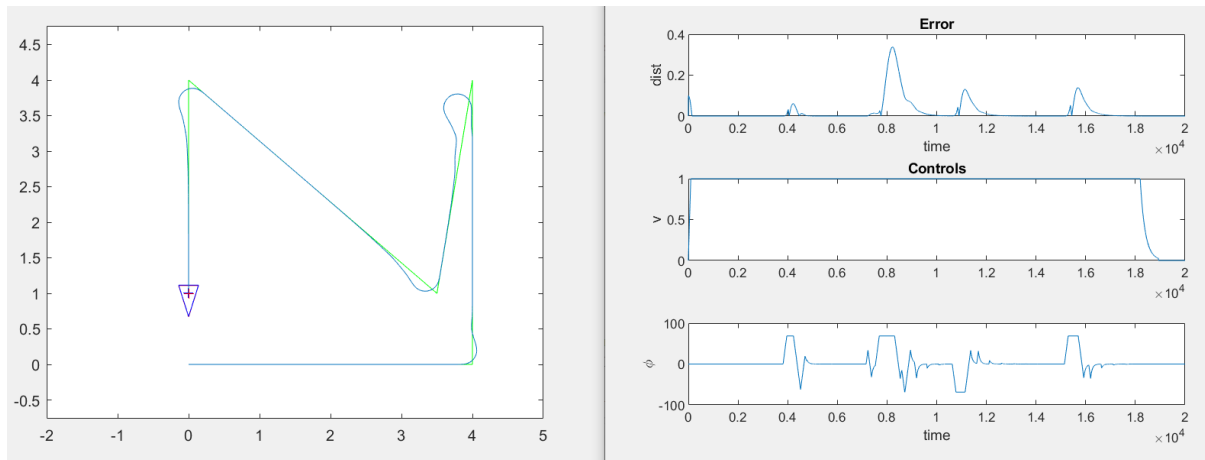


FIGURE 1: Chemin suivi par le modèle de robot bicycle sans anticipation avec un pas de discrétisation de 0.49 m, $K_p = 5$ et $K_\alpha = 7.8$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 433,0053.

plus faible que sans anticipation.

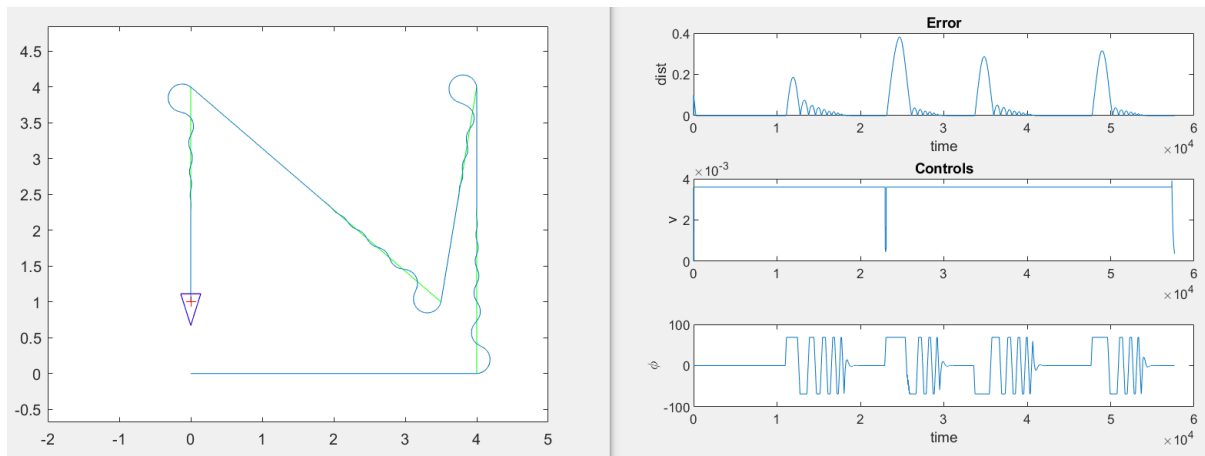


FIGURE 2: Chemin suivi par le modèle de robot bicycle avec une fenêtre d'anticipation 5 points, $K_p = 3,6$ et $K_\alpha = 10$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 2061,8577.

Ensuite, on essaye la commande avec l'anticipation avec les horizon 1 (Figure 3), 20 (Figure 4) et 100 (Figure 5) et 1000 (Figure 6). C'est-à-dire en poursuivant points vers 0,01 m, 0,2 m, 1 m et 10 m devant le robot. Afin d'améliorer la visualisation, les résultats de ces simulations ont été réunis dans le Tableau 1.

En analysant les simulations, on note que l'erreur totale n'est pas le seul indicateur à

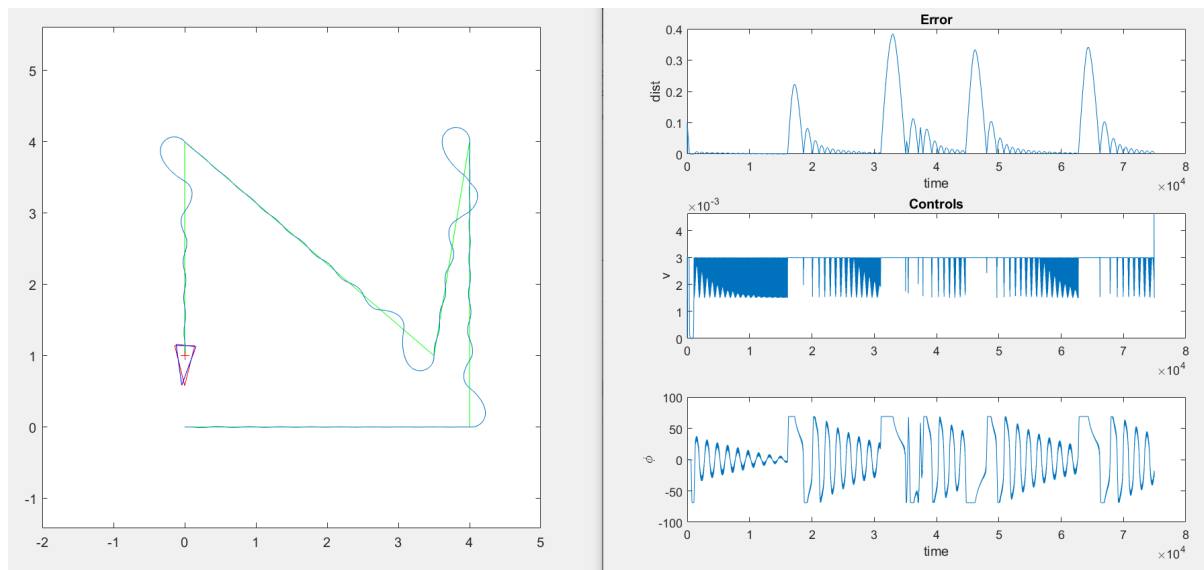


FIGURE 3: Chemin suivi par le modèle de robot bicycle avec une fenêtre d'anticipation 1 point, $K_\rho = 3$ et $K_\alpha = 0,8$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 3621,1785.

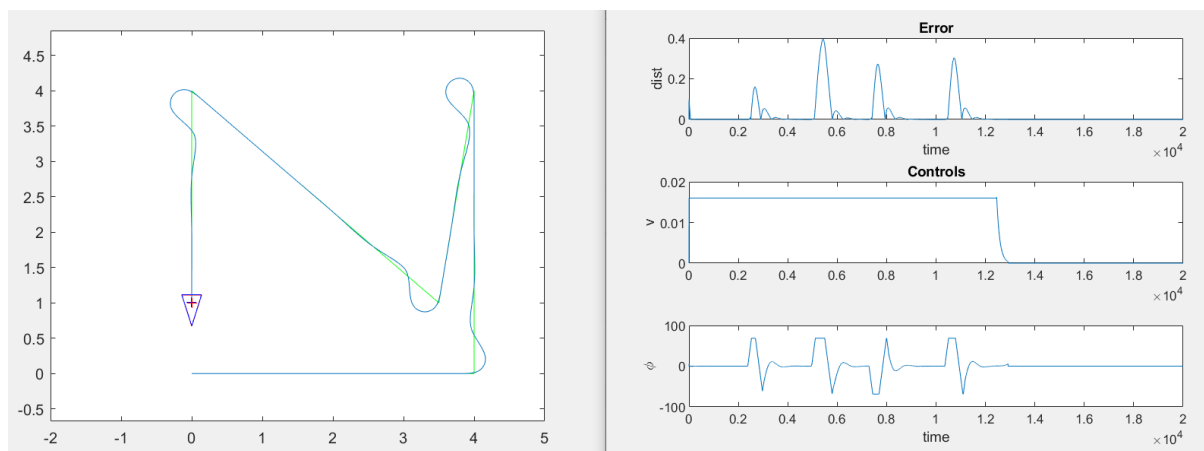


FIGURE 4: Chemin suivi par le modèle de robot bicycle avec une fenêtre d'anticipation 20 points, $K_\rho = 16$ et $K_\alpha = 2,3$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 465,6555.

utiliser parce que la trajectoire de la commande avec anticipation de 1000 fenêtres est très mauvais cependant sa erreur totale est plus petite que l'erreur des commande avec 1 et 5 fenêtres en raison du temps qu'il prend pour réaliser la tâche.

De plus, on note qu'au fur et à mesure que l'anticipation augmente la trajectoire est plus souple, le temps d'exécution est plus petit et l'erreur est plus petite aussi sauf avec 1000 points

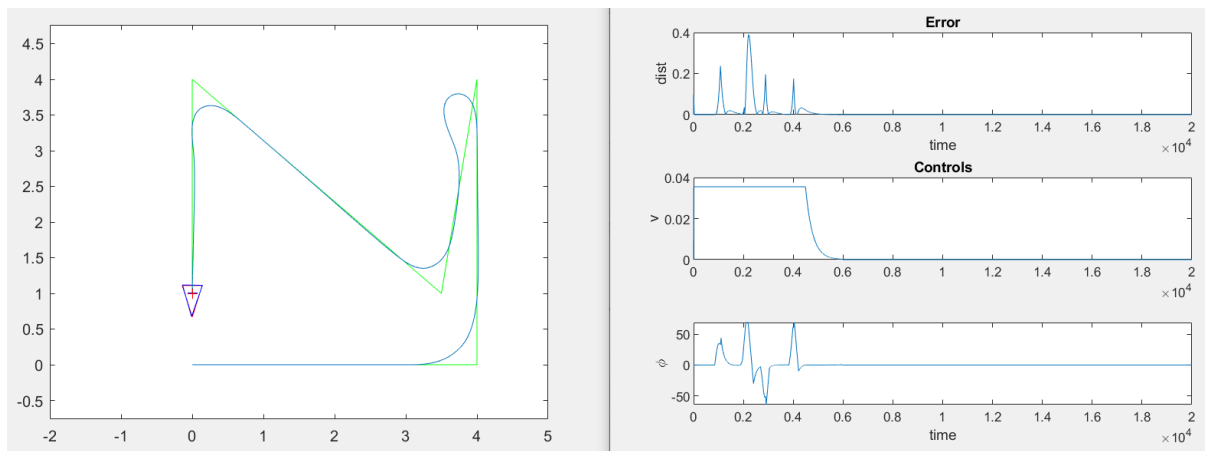


FIGURE 5: Chemin suivi par le modèle de robot bicycle avec une fenêtre d'anticipation 100 point, $K_\rho = 35,5$ et $K_\alpha = 1,7$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 188,7332.

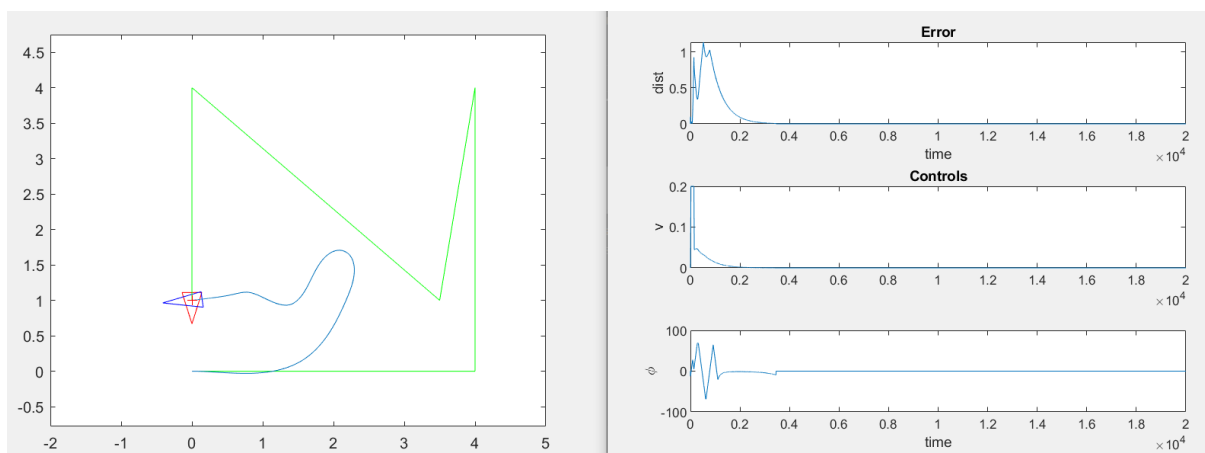


FIGURE 6: Chemin suivi par le modèle de robot bicycle avec une fenêtre d'anticipation 1000 point, $K_\rho = 200$ et $K_\alpha = 5$ (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche). Erreur totale : 1052,662.

d'anticipation à cause de la trajectoire qui triche. Ailleurs, les commandes avec anticipation trichent plus avec fenêtres d'anticipation plus grandes.

Finalement, on voudrait remarquer que cette commande avec anticipation n'est pas MPC. Dans cette commande, on n'utilise qu'un point de la trajectoire désirée plus éloigné comme référence.

Anticipation	K_ρ	K_α	Erreur totale	Temps total
1	3	0,8	3621,1785	74500
5	3,6	10	2051,5954	57350
20	16	2,3	465,6555	20000
100	35,5	1,7	188,7332	20000
1000	200	5	1052,662	20000

Tableau 1: Résultats de la commande avec anticipation d'un modèle bicycle.

Question 2 - Zone de stabilité d'une commande prédictive

En suivant l'article [H. Chen, 1998], on a effectué la vérification de la stabilité du commande d'après un point de départ. On obtient la même matrice du régulateur $K = \begin{bmatrix} 2,1180 & 2,1180 \end{bmatrix}$ et on trouve la même solution de l'équation de Lyapunov $P = \begin{bmatrix} 16,593 & 11,593 \\ 11,593 & 16,593 \end{bmatrix}$

Question 3 - Commande prédictive

Selon [H. Chen, 1998], la zone de stabilité pour $\mu = 0,5$ est la zone dedans l'ellipse de la Figure 7. Alors, on peut vérifier notre code avec les valeurs dedans cette zone.

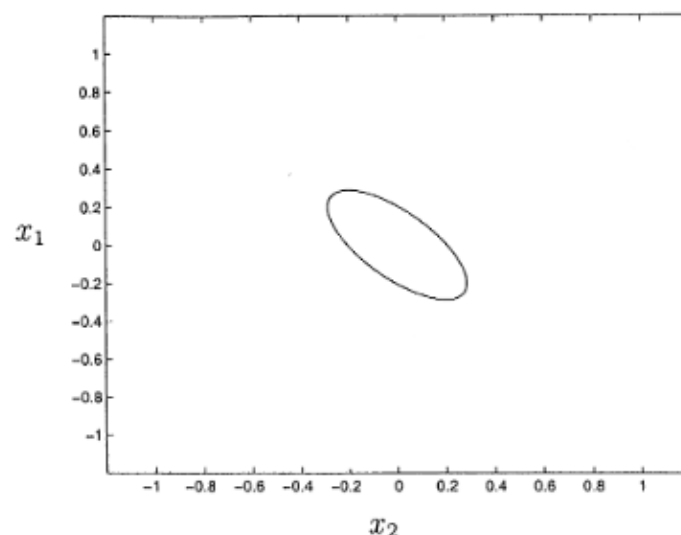


FIGURE 7: Zone de stabilité pour μ égal à 0,5 [H. Chen, 1998]

Pour vérifier si le commande trouvé est, en fait, stabilisant on essaye tourner le fichier `simulateMPC.m` avec deux valeurs dedans la zone stabilisant (Figure 7) comme on voit dans les Figures 8 et 9.

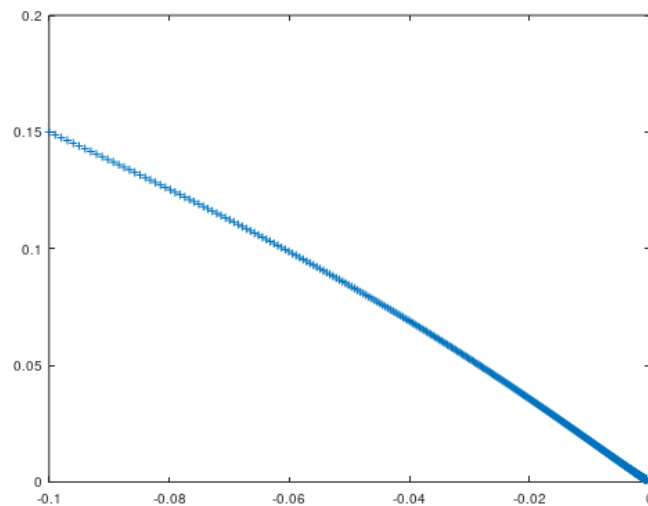


FIGURE 8: Trajectoire avec point de départ $x_1 = -0,1$ et $x_2 = 0,15$.

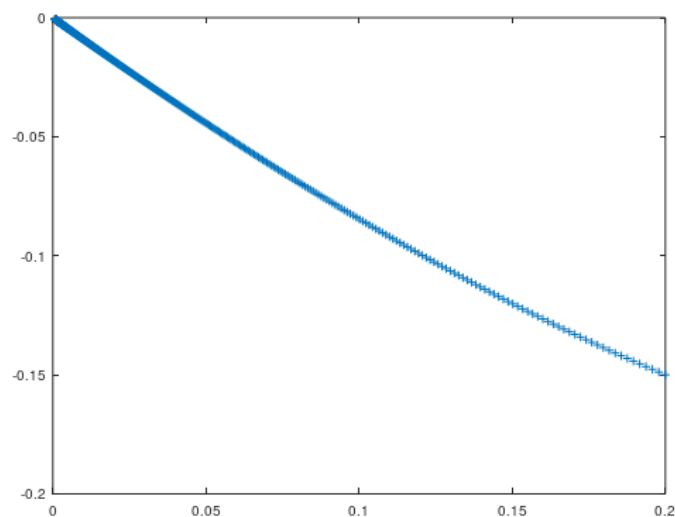


FIGURE 9: Trajectoire avec point de départ $x_1 = 0,2$ et $x_2 = 0,15$.

D'après ces trajectoires suivies, la commande MPC implémentée est stabilisant et fonctionne bien. On signale un détail important du code : en réalité, il considère toujours la matrice $K = \begin{bmatrix} -2,1180 & -2,1180 \end{bmatrix}$.

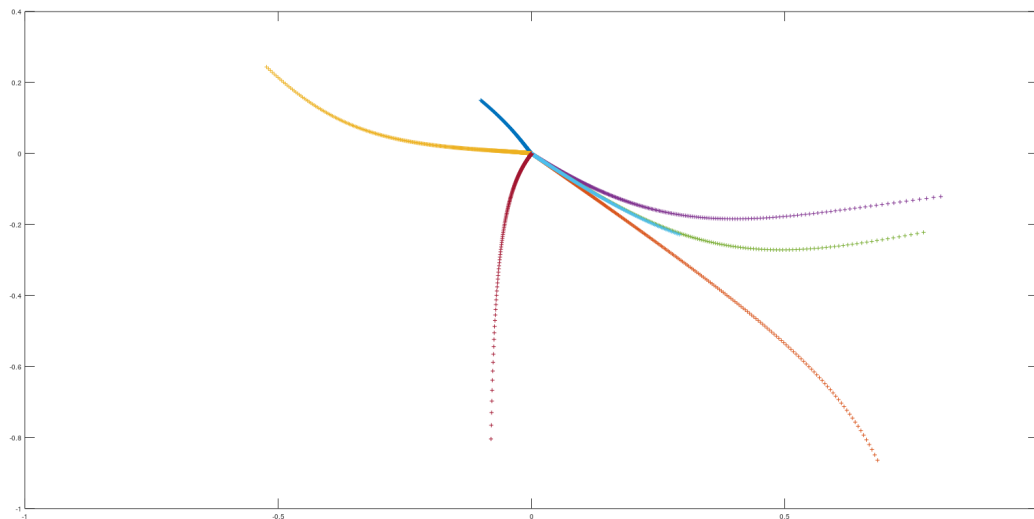


FIGURE 10: Trajectoires suivies lorsqu'on tourne le fichier testMPC.m

Bibliographie

F Allgower H. Chen. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10) :1205 – 1217, 1998. ISSN 0005-1098.